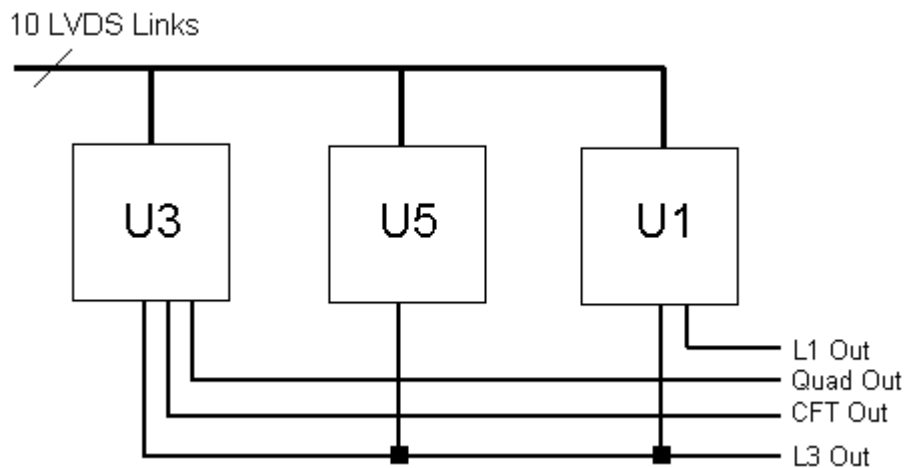


### 1 Introduction

L1 FPD equations implementation <sup>1</sup> has been conceived in order to use the three devices, FPGA <sup>2</sup>, located into the DFE Double Wide Daughter Board (DWDB). Each device will contain the equations for PU1+PU2, PD1+PD2, or D1+D2.

The first approach of this implementation requires carrying all the AFE inputs in LVDS <sup>3</sup> buses. Each DWDB <sup>4</sup> has 10 LVDS links at the input, and 7 LVDS links at the output.

Considering that the CTT Board is the one that FPD is going to use, the DWDB basic structure will look as appears in the above figure.



**Figure 1** DFE Double Wide Daughter Board

Note that the outputs, in specific cases are duplicated, or use a different kind of link. For instance, L3 Output uses two different kind of buses, one for LVDS, and one for G-Link transmission protocol <sup>5</sup>.

---

<sup>1</sup> Integration of FPD detectors into DØ electronics: Pierrick Hanlet's home page  
<hanlet@fnal.gov>

<sup>2</sup> FPGA: Field Programmable Gate Array

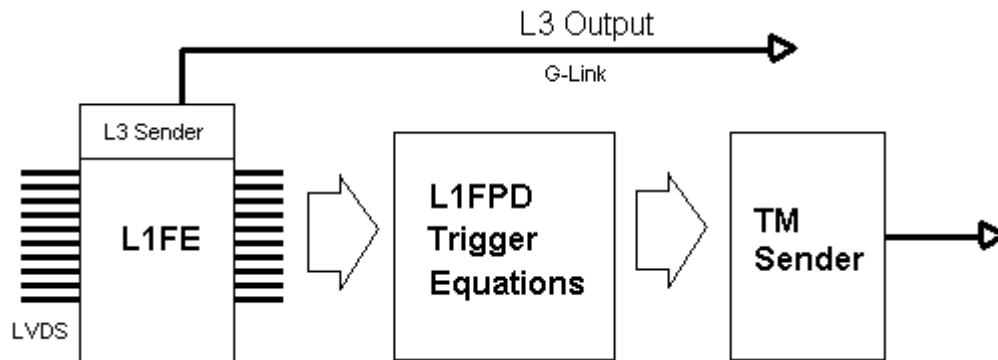
<sup>3</sup> LVDS: It is a generic multipurpose interface standard for high speed / low power data transmission.

<sup>4</sup> DWDB: For more information see Jamieson Olsen's Engineering Notes <jamieson@fnal.gov> numbers 2001-08-28a and 2001-08-28b.

<sup>5</sup> G-Link: 1Gb/sec optical link.

### 2 Global Model for each FPGA

The main idea is implementing the Mario Vaz and Wagner Carvalho's equations into a model in which they have a modular representation. It will let to do a change in a simple small part at the time that an update is required, instead of doing it in the whole code.



**Figure 2** General Descriptions

In the below table, it is possible to check the position of each piece of firmware according to the Spectrometer in which is going to be placed.

**Table 1** DFE - FPGA disposition for FPD Spectrometers

	DFE		
	1	2	3
FPGA			
U1	PU 1,2	AU 1,2	PO 1,2
U3	PD 1,2	AD 1,2	AO 1,2
U5	PI 1,2	AI 1,2	DI 1,2
	18 Links		

## 3 Proposed AFE – DFE Transmission Protocols

Proposals for the Transfer Information<sup>6</sup> between AFE Board and DFE Board. It is going to depend in how the boards and the cabling are disposed.

### 3.1 First Approach

Frame			27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00						
Header			1	1	1	HP	PM		Spectrometer										Side										L1 / L2							
F2			0	0	0	HP	U1 1:4					U1P 1:4					X1 1:4					X1P					V1					V1P				
F3			0	0	0	HP	U1 5:8					U1P 5:8					X1 5:8					X1P					V1					V1P				
F4			0	0	0	HP	U1 9:12					U1P 9:12					X1 9:12					X1P					V1					V1P				
F5			0	0	0	HP	U1 13:16					U1P 13:16					X1 13:16					X1P					V1					V1P				
F6			0	0	0	HP	U1 17:20					U1P 17:20															V1					V1P				

Figure 3 DFEA - DFE Transfer Protocol 1<sup>st</sup> Proposal

### 3.2 Second Approach

Frame																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													
-------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Figure 4 DFEA - DFE Transfer Protocol 2<sup>nd</sup> Proposal

The Spectrometers ID is showed in the above table.

Table 2 Spectrometer ID

1	0	0			PU		1	0	0			AU
0	0	1			PD		0	0	1			AD
0	1	0			PI		0	1	0			AI
1	1	1			P0		1	1	1			A0

It is important to notice that this protocol must be according with the CTTT protocols<sup>7</sup>. This will help FPD in D0 integration and Hardware debugging.

<sup>6</sup> Mario Vaz's proposal from Spring 2002

<sup>7</sup> CTT Protocols: Version 7.0 – April 5<sup>th</sup>, 2002.

### **4 Modules Description**

This design has been conceived according to the methodology Top-Bottom; this specific kind of coding uses a Top Level entity, which calls sub-blocks with hierarchy order.

#### **4.1 L1FE (L1 Front End)**

This module realizes the Synchronization function of the LVDS links. This code has been described in detail in the D0 Note number 3881 (July 18, 2001).

#### **4.2 L3 Sender**

This piece of code does the formatting and labeling of the information that will be send to L3 reading. This information will be transmitted using a G-Link protocol, and it will be arranged according to the CTT protocols.

More information about this code is available in the D0 Note number 3882 (July 2001).

#### **4.3 L1 FPD Trigger Equations**

This code has been structured in order to show an individual arrangement for each stage of the L1 Trigger process. It will be showed with more detail further on in this document.

#### **4.4 TM Sender (Trigger Manager)**

This sender is very similar to the one used by CTTT firmware design<sup>8</sup>.

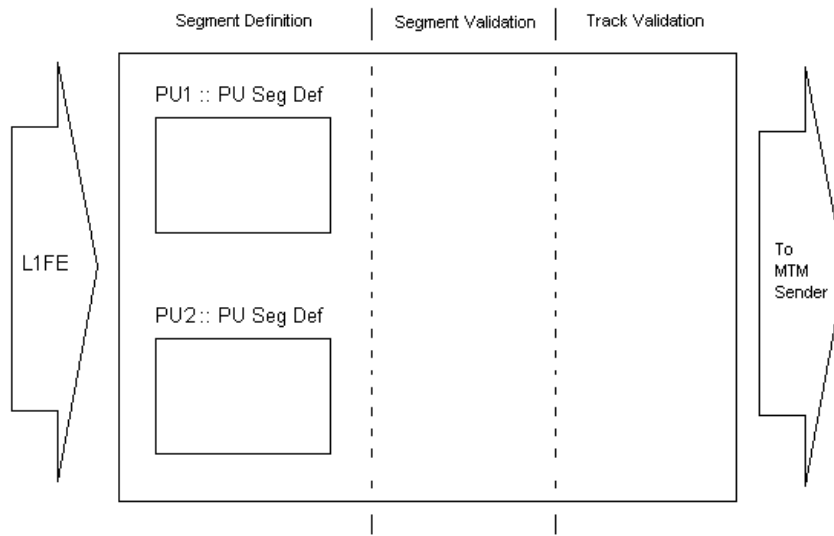
---

<sup>8</sup> CTTT Firmware: This firmware design has been made by Levan Babukhadia  
<blevan@fnal.gov>

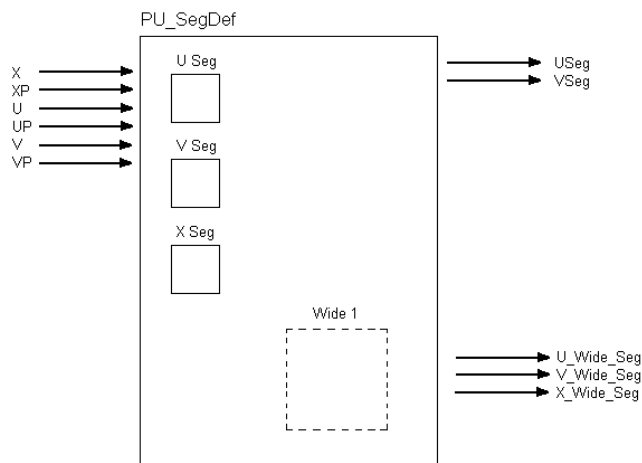
## 5 L1 FPD Trigger Equations

This module has been subdivided in small blocks, which isolate the definitions for each set of equations. According to the C++ arrangement, L1 FPD module will have three defined stages Segment Definition, Segment Validation and Track Validation<sup>9</sup>.

Furthermore, this approach is based into the Mario Vaz's first implementation<sup>10</sup> for L1FPD.



**Figure 5** L1FPD Equations Module – Top Level Entity



**Figure 6** Segment Definition Block

<sup>9</sup> C++ Code for FPD L1 Trigger Equations: Wagner Carvalho's home page

< wagner@alpha1.lafex.cbpf.br >.

<sup>10</sup> Forward Proton Detector L1 Trigger Electronics: Mario Vaz's home page <mariovaz@fnal.gov>

## 5.1 Segment Definition

These modules have a description, obtained from the C++ original generator, which uses a *generate* process<sup>11</sup> in order to create the equations into the devices. It will avoid doing the process of generation of those equations using the C compiler; also, it will give a better performance at the time in which the equations are synthesized.

In the example showed in Table 1, the C++ equations are translated to a simple VHDL *generate* process.

**Table 3** C++ to VHDL generate process

```
void seg_U (ofstream& out_u)
{
    for ( int i=1 ; i<=20; i++)
    {
        // Make U
        if (i==1)
        {
            out_u << "u_seg(" << 4*i-3 << ") <= ( not u(" << i << ") and up(" << i << ") );\n" ;
            out_u << "u_seg(" << 4*i-2 << ") <= ( u(" << i << ") and up(" << i << ") );\n" ;
            out_u << "u_seg(" << 4*i-1 << ") <= ( u(" << i << ") and not up(" << i << ") and not up(" << i+1 << ") );\n" ;
            out_u << "u_seg(" << 4*i << ") <= ( u(" << i << ") and up(" << i+1 << ") );\n" ;
        } else if (i>1 && i<20)
        {
            out_u << "u_seg(" << 4*i-3 << ") <= ( not u(" << i-1 << ") and not u(" << i << ") and up(" << i << ") );\n" ;
            out_u << "u_seg(" << 4*i-2 << ") <= ( u(" << i << ") and up(" << i << ") );\n" ;
            out_u << "u_seg(" << 4*i-1 << ") <= ( u(" << i << ") and not up(" << i << ") and not up(" << i+1 << ") );\n" ;
            out_u << "u_seg(" << 4*i << ") <= ( u(" << i << ") and up(" << i+1 << ") );\n" ;
        } else if (i==20)
        {
            out_u << "u_seg(" << 4*i-3 << ") <= ( not u(" << i-1 << ") and not u(" << i << ") and up(" << i << ") );\n" ;
            out_u << "u_seg(" << 4*i-2 << ") <= ( u(" << i << ") and up(" << i << ") );\n" ;
            out_u << "u_seg(" << 4*i-1 << ") <= ( u(" << i << ") and not up(" << i << ") );\n" ;
        }
    }
}
```

```
USeg1: for i in 1 to 20 generate
    USeg1a: if (i=1) generate
        u_seg(4*i-3) <= ( not u(i) and up(i) );
        u_seg(4*i-2) <= ( u(i) and up(i) );
        u_seg(4*i-1) <= ( u(i) and not up(i) and not up(i+1) );
        u_seg(4*i) <= ( u(i) and up(i+1) );
    end generate;
    USeg1b: if ( 20 > i and i > 1) generate
        u_seg(4*i-3) <= ( not u(i-1) and not u(i) and up(i) );
        u_seg(4*i-2) <= ( u(i) and up(i) );
        u_seg(4*i-1) <= ( u(i) and not up(i) and not up(i+1) );
        u_seg(4*i) <= ( u(i) and up(i+1) );
    end generate;
    USeg1c: if (i=20) generate
        u_seg(4*i-3) <= ( not u(i-1) and not u(i) and up(i) );
        u_seg(4*i-2) <= ( u(i) and up(i) );
        u_seg(4*i-1) <= ( u(i) and not up(i) );
    end generate;
end generate;
```

<sup>11</sup> VHDL generate process: Documentation in Xilinx support home page.

### 6 Verification and Testing

In order to do the testing of this design is required to reformat the provided C++ Test Vectors. These vectors must agree with the VHDL Test Bench<sup>12</sup> format, which put rows of 196 bits for each LVDS link.

These vectors are required in a second presentation. The Hardware debugging, in principal, is based in the Data Pump (DP)<sup>13</sup> verification tool. DP provides communication between a normal PC, via Parallel Port, and the DFE boards in the Combined Test Stand (CTS). The new format must have a Hexadecimal convention for representing each of the inputs that will be send them to DP through a Data Sheet interface.

---

<sup>12</sup> Test Bench: For more information look for D0 Note number 3871 (June 12, 2001).

<sup>13</sup> Data Pump: Jamieson Olsen's Home page <jamieson@fnal.gov>. Also, see his engineering note number "2001-08-28b".